Computer implementations on light-scattering methods

Antti Penttilä

2016

Methods and codes that we deal here

- Mie for spherical particles

- T-matrix for rotationally symmetric particles

- Multiple sphere T-matrix for collection of spheres

- Discrete dipole approximation for any shape

Hands-on work with the codes

Mie code

- Based on spherical vector wave function series where the fields are represented. Series coefficients are solved.
- Applicable to spheres and coated spheres
- When the size parameter of the sphere gets large, also the required truncation number of the infinite series gets large, and the computations slow
  - In practice, this happens when the sphere is already in the geometric optics regime
- Implementations: BHMIE (Fortran, IDL, Matlab, C, Python – check Wikipedia for Mie codes), MatScat (for Matlab, google the link)

Mie task

- Sphere with size parameter $x = kr = 2$, ice material with refractive index $m=1.31+i\,0.001$
- Compute scattering phase function (P_{11}-element of the Mueller matrix) and degree of linear polarization ($-100\%\ P_{21}/P_{11}$ vs. phase angle)
- Plot scattering efficiency Q_{sca} [$=C_{sca}/(\pi x^2)$] as a function of size x, varying x from 0.5 to 20. Why Q_{sca} can be more than one?

T-matrix code

- Also based on vector spherical wave function basis and coefficients for the functions
- Transition (T) –matrix can be solved for all shapes, but is computationally efficient for rotationally symmetric particles
- Implementation by Mishchenko, see http://www.giss.nasa.gov/staff/mmishchenko/t_matrix.html and "**Double-precision** *T*-matrix code for **randomly** oriented nonspherical particles"
- Good with moderate size parameters (max. some tens) and moderate aspect ratios (<4)
- Old-fashioned Fortran77-code, all input variables hard-coded in the source file, so altering them requires compiling

- Oblate spheroid particle with aspect ratio 2, x=2 (volume-equivalent size parameter), m=1.31+i0.001
- Compute scattering phase function (P_{11}-element of the Mueller matrix) and degree of linear polarization (-100% P_{21}/P_{11} vs. phase angle), compare to sphere

Multiple sphere T-matrix code

- Based on vector spherical wave functions and translation to joint origin
- MSTM, applicable to sphere clusters and spheres inside spheres, sphere surfaces cannot cross each other
- Works with hundreds of spheres (moderate size parameters, not too sparse aggregate) or parallel version with thousands of spheres
- Implementation by Mackowski, http://www.eng.auburn.edu/~dmckwski/scatcodes/, Fortran 90 code with input file for parameters

MSTM task

- Create aggregate/cloud with ten x=2, m=1.31+i0.001 spheres. You can place the spheres as you wish
- Compute phase function and degree of linear polarization in random orientation setup

Discrete dipole approximation

- Based on volume-integral-equation solution for Maxwell equations in the approximation where scatterer(s) are replaced with small dipole elements. For efficiency (FFT-acceleration) the dipoles are located in rectangular grid
- Applicable for targets with size parameter up to 100, moderate refractive index (below 2 or 3 preferably)
  - Memory requirements grow with size, parallel computing needed for large sizes
- Few implementations, most popular are ADDA (Yurkin, C, https://github.com/adda-team/adda) and DDSCAT (Braine, Fortran 90, http://ddscat.wikidot.com/)

DDA task

- Same as Mie, sphere with size parameter $x = kr = 2$, ice material with refractive index m=1.31+i 0.001
- Compute scattering phase function (P_{11}-element of the Mueller matrix) and degree of linear polarization (-100% P_{21}/P_{11} vs. phase angle)
- Compare to Mie solution
- Sphere is among predefined shapes so you don't need to create the shape discretization yourself
  - Discretation accuracy (i.e., dipole size, dipoles per lambda) an important parameter. About ten dipoles at least per wavelength